Integrated Planning and Execution for a Self-Reliant Mars Rover

Steve Schaffer and **Joseph Russino** and **Vincent Wong** and **Heather Justice** and **Daniel Gaines**
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, California 91109
firstname.lastname@jpl.nasa.gov

Abstract

Planetary rovers exploring the surface of Mars face a challenging operational environment that requires close cooperation between deliberative planning and behavioral execution in order to most efficiently leverage the robot's capabilities into science value returned to earth. The Self-Reliant Rovers project envisions future rover missions that require only occasional high-level direction from human controllers to successfully conduct detailed in-situ studies of its Martian environs. To achieve this high degree of autonomy, this work leverages a spectrum of planning and execution techniques that allow the rover to respond appropriately to both opportunity and adversity it encounters. Small perturbations are accommodated at first by behavioral adaptation, with more and more extensive disruptions handled in turn by executive administration of plan flexibility, heuristic-guided plan repair strategies, and finally comprehensive replanning from science campaign goals. The integrated system has been deployed and tested on a terrestrial rover in an environment and under scenarios that anticipate those faced by future Mars rovers. This paper recounts complexities of planning and execution coordination faced in the rover domain and the practical solutions employed to address them. Particular emphasis is given to lessons from the field and foibles ripe for remedy by future advances in planning and execution research.

Introduction

Current planetary rover operations for Mars Science Laboratory (Vasavada et al. 2014) exert a significant daily workload on mission operations staff. Within one work shift, scientists and engineers must interpret downlinked data, reevaluate overarching mission goals, and then synthesize a responsive activity plan and detailed command sequence covering the next planning period. The nominal single martian day planning period allows the team to maintain a high level of productivity via timely manual response to execution eventualities, but there is markedly diminished efficiency when plans must span multiple days (Gaines et al. 2017a) (e.g. due to off-sync relay passes or institutional holidays.) Future Mars surface missions are anticipated to have even more frequent and extensive lapses in regular communication due to turnover in available relay craft (Edwards et al. 2014), and will thus require more robustly integrated onboard planning and execution to sustain productivity absent prompt human feedback. Even with regular communication, the productivity of current rover missions has benefited greatly from onboard collaboration of imaging activity execution with automated data analysis and goal selection (Francis et al. 2017).

The Self-Reliant Rovers architecture seeks to further extend onboard planning and executive capabilities so that future rover missions can continue productive scientific inquiry without constant micromanagement by human controllers. The SRR architecture was borne out of thorough study of current rover operations and the potential efficiencies attainable by increased onboard autonomy (Gaines et al. 2016). Under the SRR architecture, rover objectives can be expressed as high-level campaign intents rather than meticulously assembled daily activity plans (Gaines et al. 2017b). For example, a scientist might request detailed imagery of any quartz veins detected during a walk about of a boulder field, while another may request recurring atmospheric opacity measurements every day at noon. Engineers might specify mandatory relay communication passes along with required battery reserves at the end of the planning period. The onboard planning and executive functions can leverage the discretion entailed in such conceptual guidance to respond directly to unpredictable outcomes and continue exploration during communication gaps.

This paper outlines challenges to effectively integrated planning and execution within the Mars rover domain, along with practical techniques employed within and between SRR components to achieve the envisioned level of rover mission autonomy. Initial results on a terrestrial rover test bed are presented, and fertile areas for future enhancements to the integrated system are described.

Approach

The Self-Reliant Rover system is designed within the context of the Jet Propulsion Laboratory flight software architecture (Weiss 2013) and incorporates a tiered robotic control architecture. At the highest level, scientists and engineers use the MSLICE graphical interface (Powell et al. 2009) to construct both general campaign objectives and detailed constraints that will guide the system's behavior. The goals are then transmitted to an onboard optimizing activity planner, CASPER (Chien et al. 2000), which assembles and maintains a comprehensive working schedule for

Figure 1: The Athena rover imaging a science target during a simulated Mars surface mission.

the rover that fulfills the requests while respecting vehicle safety limits. Activities are then dispatched from this schedule to a purpose-built reinterpretation of the MEXEC state-based executive (Verma et al. 2017) that oversees their coordinated execution and reports back on their ongoing status. The executive calls upon task-oriented behavior components, and those in turn refer to low-level functional components, that together accomplish specific robotic tasks such as locomotion, imagery, and data analysis. Many components are reused from the CLARAty library of portable robotic software modules (Volpe et al. 2001). Channelized robot state and resource updates are published by the cognizant components, and may be subscribed to by any other interested component. Component encapsulation and intercommunication are provided by the ROS framework (Quigley et al. 2009).

Planning Updated high-level objectives are provided to the onboard planner by the ground operations team on an intermittent schedule, but special onboard system nodes are also empowered to submit new constraints and goals to the planner. Using guidance from scientists, the automated science data analysis component recognizes features in acquired imagery that may warrant further study, such as boundaries between geologic deposits. The analysis component calculates the location of the new targets in the environment using camera model and rover position metadata that is attached to each image, and constitutes new goals for follow-up science activities based on a template provided by the science team. The template allows humans to bound the self-directed goal behavior by specifying details such maximum number of follow-ups, total priority/utility to assign, and association to broader science campaigns. The new goals are submitted to the planner and are integrated into future decisions along with the rest of the pending goals and constraints. Similarly, a vehicle health management component monitors ongoing rover performance and may respond to anomalous trends by imposing tighter safe operation limits or calling for diagnostic activities. The planner remains the arbiter of when to undertake new activities since it has a broader picture of the rover's future plan, including upcoming critical activities such as communication passes.

A key feature of the system is the onboard planner's flexibility to either heuristically repair the existing plan in the face of small perturbations or to regenerate a new plan forward from the as-executed stem in the case of more extensive disruptions. Replanning is not especially prohibitive ($<$1 minute), thanks to the efficient domain-specific multi-threaded best-first branch-and-bound anytime path/plan optimization algorithm employed, but it is still intensive enough that full replanning cannot be performed every update cycle. Accordingly, the criteria under which full replanning is invoked are relatively liberally construed as those which have reasonable probability of meaningfully changing the the path among goal locations or the science and engineering activities conducted along the route. A incoming batch of new goals thus always triggers replanning, as does failure of an activity declared by the executive. Replanning is also invoked when updates to rover state or resource levels propagate into predicted conflicts in the future plan, for example due to a late-running motor preheating task. A more subtle trigger examines the unused resource and time margins, and calls for replanning if the excess overtakes some threshold, as might be the case after a series of better-than-expected executions.

Because they are less likely to result in structural changes to the plan, minor updates that reach the planner are handled by rapid plan repair heuristics. Resource and state updates are posted to the plan at the time they are received, with fast re-prediction propagating the timeline's expected future values and checking for any conflicts with upcoming activities or constraints. Importantly, timeline updates that do not immediately trigger predicted conflicts are still collected and posted to the plan to inform future predictions, which may finally rise to a conflict only after several small discrepancies are recorded. When activities end early, a dynamic packing heuristic adjusts future action start times as close to the present as avoids inducing any plan conflicts. Any associated activities, such as mechanism preheating, are also moved forward. This results in filling unused blocks of time following hastened activities, while leaving absolute-timed activities such as communication passes at their proper time.

Late running actions cannot be accommodated in the same manner in general; by the time a preceding action is known to be late, the subsequent activity may have already been dispatched to the executive. The main loop of the planner must dispatch activities well enough in advance so that no start times are missed during its full iteration duration, which may be extensive in cases where full replanning is invoked. The current SRR system inherits a fixed duration commit window that the anytime path solver algorithm is obliged to respect: activities within the window must not be modified since they were already dispatched to the executive, and the solver must return control to the main loop by the end of the window, regardless of its solution progress. As on online algorithm, the solver is able to submit the best plan so far at any break point, and is also able to restart from its previous partial solution state. Deconflicting dispatched activities when one runs past its planned end time becomes the responsibility of the executive, but is informed by the planner's model of the individual activity state and resource constraints, which are also passed down.

Execution When the designated start time for a dispatched action arrives, the executive first checks the constraints from the planner before initiating the activity. If all of the constraints are not met (for example, a late running panorama is still using the rover mast also needed for a targeted image), the subsequent activity start is held back by the executive until either the constraints are satisfied or a delay threshold is reached. The planner may select different delay thresholds for each dispatched activity instance in order to communicate contextual start time flexibility from the plan constraints. The eventual activity start and end times are reported by the executive to the planner and other interested components to ensure accurate resource modeling and vehicle health assessment. This approach allows the executive to locally handle small delays that do not have a large impact on the plan structure, but in a way that is consistent with the planner's expectations of activity preconditions.

The executive contains another layer of precondition checking for safety purposes: a hard-coded table from the system engineering team that delineates which activities are safe to execute concurrently with each other and specific rover states. For example the drive action might require that the rover arm is in the stowed state and that no other activity is using the navigation cameras. Before finally initiating any new activity that is otherwise cleared for execution, the executive checks it and any other ongoing activities against this compatibility table. A failure to pass the safety check results in rejection of the new activity by the executive and failure notification being sent to the planner. This is different than for planner supplied constraints, which merely delay execution in anticipation of imminent state updates. A safety check rejection indicates that the planner is direly unaware of the current execution conditions, or fails to model an important system safety rule.

After the executive initiates an activity, it makes calls to the required lower-level behavioral task components and then continues monitoring their ongoing progress with a small state machine. Simple tasks just return a status message on their completion, in which case the executive notes the end time of the activity, clears the state machine, and forwards the result (success or failure) to the planner. More sophisticated activities include additional monitoring of behavior start up and progress reports, enforcement of rover state conditions throughout the activity execution (akin to the precondition safety checks), and activity termination criteria monitoring. In the event a monitored in-condition of the activity is violated, the behavior is signaled to immediately abort in order to prevent vehicle damage, and the planner is notified of the activity's failure. By default, activities are also checked against their planned end-time, and any overruns past a chosen threshold trigger the same executive abort response. This means that the planner model of action durations must be pessimistically long, though the resultant plan inefficiencies are largely recovered during execution by the dynamic packing and replanning features of the overall system.

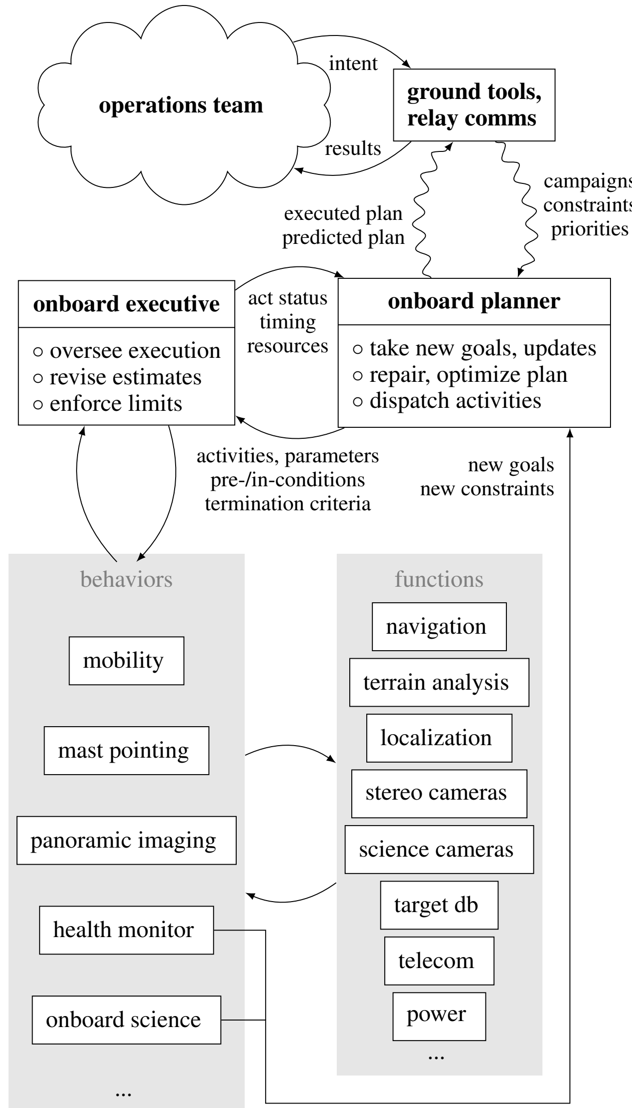

Figure 2: Self-Reliant Rover architecture overview. Scientists and engineers provide high level objectives, which are optimized in-situ by an onboard planner. The resultant activities are managed by an executive that understand constraints and flexibility in the plan. Individual behaviors invoke even lower-level modular functions. The executive and planner cooperate to respond to unexpected outcomes, changing resource estimates, and even new goals and constraints borne out through execution.

Driving The drive behavior component includes many additional features to allow close cooperation between the planner and executive in meeting the challenges posed by the rugged Martian terrain. Drive activities serve as connectors between all of the science activity locations in the plan, but are highly complex planning notions in their own right, as they continuously vary the rover's position and resource use over an extended period. Plausible science or engineering campaigns requested to recur e.g. every 50 meters of drive distance, every 10 meters of elevation change, or every 3 hours of elapsed time are all profoundly impacted by the precise drives in a plan. Further interactions are mediated by the stationary state requirement of some activities such as fixed communication passes, regenerative sleeping, and almost all science activities. On top of this, drives represent the most significant execution uncertainty for planetary rovers due their highly-coupled interactions with the unexplored alien environment.

Overview orbital imagery can only give rough clues about the surfaces and obstacles that will be encountered along different possible drive routes. Rovers have thus long relied on onboard stereo vision, obstacle avoidance, and visual odometry embedded in the locomotion services (Goldberg, Maimone, and Matthies 2002), and SRR continues this tradition. The unpredictable diversions around obstacles can lead to significant discrepancies with estimated arrival times, as well as drawing the rover off of its expected path. The primary impact of driving delays is on the scheduling of subsequent activities, either because they required the rover be at a specific location or to be stationary, but there are also downstream effects on rover resource and state predictions as well. In case of diversions significantly off of the planned course, it may also become appropriate to adjust the plan to accomplish other goals along the detour route before resuming the initial drive. In extreme cases, the system may have to accept the eventuality that a selected target destination is really unreachable.

The constraint-based task execution strategy discussed above can accommodate some of the drive dependencies, for example by providing a location precondition on targeted science activities. This raises a question about how physical rover positions correspond to locations specified in a campaign request, which is answered jointly by the locomotion engine itself and a target association component. When determining location satisfaction, the associator takes into account additional details from the campaign such as permissible instrument range to intended target, allowed rover bearings (e.g. for illumination reasons), and instrument field of view limits. The locomotion engine has been augmented to accept the arguments when requesting a drive as well, whether those drives are for immediate execution or for hypothetical evaluation during goal planning. Using the same locomotion engine at plan and execution time ensures that the planner benefits from the latest obstacle maps and thus has a more accurate prediction of the drive behavior.

In addition to unpredictable obstacle environments, drive actions may be subject to further uncertainty due to the driving surface. The substrate texture can change abruptly from hard rock outcrop, to compacted soil, to loose sandy ridges, each with very different wheel slip characteristics. Furthermore, the slope of the terrain and the rover's approach aspect also impact drivability. Certain combinations of terrain features even amount to mission-ending wheel trap hazards that must be fastidiously avoided. Terrain classification and slip aware navigation components were incorporated into SRR to address these uncertainties (Rothrock et al. 2016). Similar to obstacle avoidance, imagery collected during drives is used to classify different textured soil types and slope inclines around the rover into different cost categories. The costs are overlaid onto the constantly updating navigation map used by the locomotion engine so that it can update its routes to avoid dangers and make the fastest progress to the goal.

Drive Termination When the locomotor updates its route during execution, it diverges from the time and distance estimates predicted at plan time. The locomotor posts progress reports that include revised estimates, which are then used by the executive to predict the likelihood of success or failure by the planned end time. Rather than waiting until that end time to post a failure to reach a target location, the executive is empowered to abort the drive early when revised estimates exceed the allotted time frame by some margin. This minimizes wasted driving effort in difficult terrain by allowing more immediate replanning with new map data. It also indirectly triggers replanning on large diversions from the expected route, since such diversions are likely to induce large changes in duration estimates.

The executive is also able to terminate drive actions early in order to meet campaign objectives based on rover states such as distance driven or time of day. This is necessary since an initially planned drive action may end up driving further or taking longer than expected, so much so that the next instance of a recurring campaign activity should be invoked. For example, if the planner must accommodate an image request every 50 meters driven, then an upward revision from 40 to 60 meters estimated drive distance requires stopping for an extra intermediate image. To accommodate this scenario, the planner looks ahead of each drive it dispatches to find recurring campaign goals may need to intercede in the drive, and attaches those campaign criteria to the drive as additional termination conditions. The executive monitors these termination conditions (e.g. odometer reading of 50 meters, or specific time of day) and aborts the drive behavior when they are met. Rather than reporting outright failure of the drive, the executive reports which termination condition stopped the drive and the actual location different from intended target location. The planner nevertheless interprets the unexpected stop as an inconsistency in its current plan, and so invokes replanning, which will likely insert the relevant campaign activity followed by a completion of the initial drive. The rover states to which such termination criteria are attached should be carefully selected to avoid ambiguities due to partial drives; for example specific odometer or clock readings should be used rather than distance or time driven from the start of a drive segment. This allows the criteria to be applied uniformly across drive segments rather than recomputed relative to each leg.

Results

The SRR system was demonstrated on the JPL Athena rover within a mission scenario that explores the JPL mini-Mars Yard robotic testing environment. The primary science objective was to characterize the rock outcrop materials embedded in the sandy soil using the rover's mast-mounted

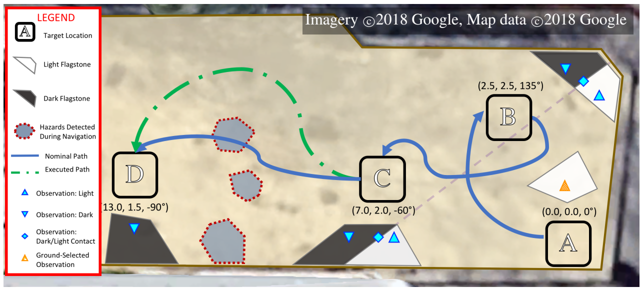

Figure 3: Overview of simulated mission area. Operator inputs include a specific target selection (orange) near starting area A along with only high-level campaign guidance for areas B, C, and D. Automated science analysis injects additional targets (cyan) during execution. The initial planned route (blue) is dynamically adjusted (green) to avoid unanticipated terrain hazards (red).

cameras. The simulated mission spans a period of limited communication with operators, so the rover must operate almost entirely autonomously in order to remain productive toward its high-level goals.

Figure 3 shows the overhead layout of the mission area, as might be available to mission planners from orbital imagery. The operations team selects several regions of interest (indicated by letters) from this coarse data, but is unable to identify specific targets or terrain obstacles beyond a few meters from the rover. They then construct a goal for each target area that entails driving to a specified vantage point, acquiring a contextual wide-angle image, and then running the automated science algorithms. The planner will stitches these goals together in an optimal drive ordering that achieves as many as possible. The scientists also create campaign goals for the desired follow-up outcrop observations in each area, including templates for goals automatically generated by the onboard science analysis. The planner and automated science cooperate to identify the best candidate targets to include in the plan so as to maximize expected utility score. In this demonstration scenario, campaigns request follow-up mast camera imaging of the 2-5 best outcrop specimens in each category at each location. Several additional relevant campaign types were demonstrated in separate scenarios. For example, the operators can specify ongoing temporal periodic campaigns such as visual atmospheric opacity (τ) measurements every 20$\pm$2 minutes. Mandatory downlink relay communication passes can also be enforced at specific times in the schedule, representing a exogenous orbiter overflights.

All of the various goals are provided to the rover at its morning communication pass at the start of the mission scenario. Thereupon, the onboard planner generates a plan to image the specifically requested target near A, and then travel in turn to B, C, and D to conduct survey observations (fig.4, top, and fig.3, blue path). The plan adheres to all rover resource limits (such as battery energy and data volume), as well as incorporating any required heating (such as needed for instruments or mobility mechanisms). The actual

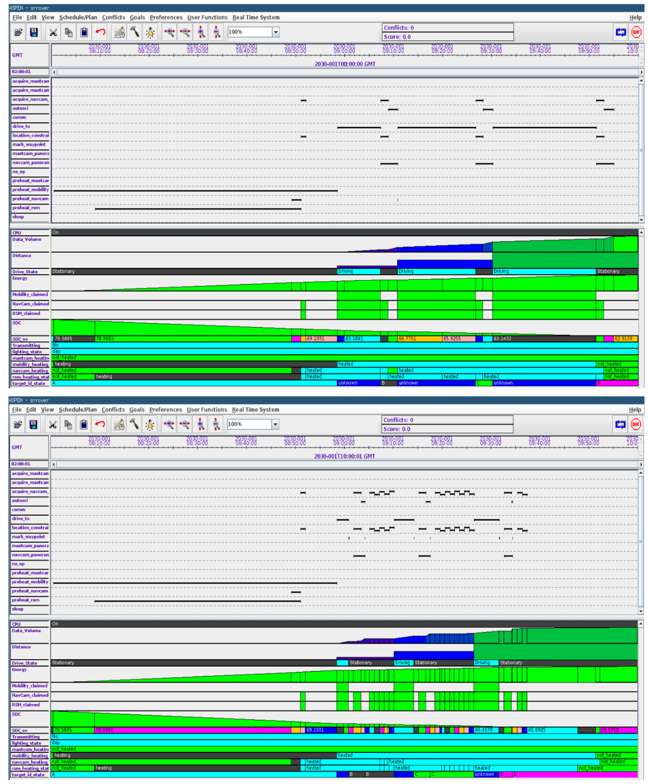

Figure 4: Initial generated plan and final as-executed plan for the simulated mission scenario. Many new targeted science goals are suggested at run-time by automated image analysis and then integrated into the schedule in service of science campaigns. Drive estimates are also updated during execution, thus correcting initial approximations.

path driven by the rover undergoes refinement by the onboard terrain classification and autonomous navigation so as to best avoid obstacles along the planned route (fig.3, green path). Diversion delays and expeditious travel cause minor perturbations to the plan, which are accommodated by the dynamic packing plan heuristic.

On arriving at B, and later C, the rover acquires the requested contextual images and analyzes them using the onboard science detectors, which in turn identify flagstone outcrops for follow-up imaging. The selected targets are then automatically injected as new goals into the planner campaigns, and a replanning cycle is initiated. The planner's updated solution includes each of the newly suggested observations, which are duly collected before proceeding to the next area.

Upon driving toward D, the rover's automated terrain classification identifies a major obstacle, and the navigation system must divert significantly. The planner incorporates updated drive estimates from the navigation engine to ensure that the plan can accommodate the delay without conflict. After planning a safe path around the observed obstacles and eventually reaching D, the system once again identifies flagstone features and conducts the requested follow-up observation. At this point the simulated mission ends.

As seen in the final plan (fig.4, bottom), the productivity benefits of additional onboard rover autonomy are evident even within the limited scope of this demonstration scenario. Traditional operations would have accomplished just one initial outcrop observation and a first drive. The combined autonomy of the SRR system produced three survey panorama images throughout the mission area, toured several unexpectedly difficult terrain routes, and accrued fifteen additional targeted outcrop observations. The scenario successfully demonstrated the mission productivity benefits of integrated planning and execution within the SRR architecture.

Future Work

A consistent decision framework for when each class of plan modification is warranted would help theoretically ground the ad hoc assignment of replanning versus repair triggers. Such a framework would likely consider a balance between the maximum or estimated utility payoff of invoking replanning versus its opportunity cost, measured both in terms of computation time as well as operational costs of disrupting an existing plan that human operators may have a stake in. Useful clues to the potential payoff are likely available as an ancillary product of the planner's branch-and-bound algorithm. New goals with large expected rewards would thus drive replanning when they became available, while several smaller adjustments would have to accumulate before warranting a replan.

The current system operates without reserving resources or time for follow-up science goals anticipated for each automated science analysis run, instead depending on agile replanning to fit in the new goals when they appear. Other possible approaches where only given cursory evaluation and deserve further study. One alternate approach is to use a placeholder activity that represents the budget of time and resources that are allotted to automatic science goals, and then decrement that reserve for each actually planned follow-up. The reserve could be over the entire planning period or attached to each separate automated science activity. Planning for the automated science analysis follow up observations also faces a classic dilemma of exploitation versus exploration. Since new nearby goals are injected following the initial image analysis, there is a tendency for the rover to get caught up in examining that first location and defer subsequent targets that could have even more valuable follow ups. Budgeting time and resources for each analysis target will help reduce the early over-exploitation, but really solving the problem requires proper incorporation of some exploration metric into the plan scoring heuristic itself. Such a metric is counter-intuitive to normal path planning since it requires driving the entire walkabout distance first to do initial analysis, and then driving it again to revisit the most interesting targets.

Tighter collaboration on activity duration adjustment is possible between the planner and executive. This would reduce the need to consistently overestimate activity durations within the planner model and the associated inefficiencies, as well as reducing the occurrence of executive delay holds on subsequent activities committed early. The planner could evaluate the range of conflict-free durations for each dispatched activity and pass along those bounds as a end time flexibility, rather than enforcing the single predicted end time as a hard limit on execution. The executive could manage the flexibility to extend activities that need just a little more time for completion without worry of damaging the plan. Alternately, the executive could pose requests for activity extensions to the planner as soon as it received revised estimates from the behaviors. The planner could then perform a hypothetical planning cycle with the extended activity and report back to the executive whether the extension is granted.

Additional integration between the locomotion components and the planner would also benefit the system. In particular, the terrain classification and obstacle detection systems are constantly improving their map of the environment, but the planner only benefits from those improvements when full replanning is invoked. Until then, the planner relies on its previously cached drive estimate responses. Instead, it may be worthwhile for the locomotion components to track which estimates the planner is currently using, and to transmit revisions to those estimates when optimal routes change by some threshold. The planner also only indirectly recognizes when the driven route has diverged significantly from the initial path via changing drive time estimates. This means that the planner is slow to respond to opportunities nearby the diverted route, perhaps even missing them completely. If the planner, executive, and locomotor collaborated on the actual path geometry (rather than just summary estimates), the system would be able to capture such detour opportunities.

Many integration hurdles could be overcome if the planner and executive shared access to the same plan. For example, the serialization of planner constraints and flexibility could be avoided, as could the convolutions of advance dispatch. However, it would require great care to ensure that ongoing execution updates could successfully interleave with the planner during active replanning. Even without such drastic merging, the commit window scheme could be improved to increase planning flexibility and reduce reliance on executive delay holds. For example, the commit window could be dynamically sized depending on execution conditions: narrow when only minor changes are being posted, but wider when full replanning is called for.

There are several interesting issues to tackle regarding integration of mission planning on Earth and with the onboard planning and execution. Because of the communication delays involved, the human planners are always operating with an old snapshot of what the rover had accomplished so far at the time of downlink. The onboard planner also communicates its then-current plan for the rest of the period so that human operators have a concept of what goals might be achieved by the time their new requests would arrive. However, the onboard planner may diverge from that plan for various reasons, meaning operators must not rely on any specific future chain of events. A balance may be struck by assigning probabilities to possible futures and expressing new goals either independent of unconfirmed actions or explicitly conditional on them. When the rover receives goals updates during an uplink pass, it must also carefully disposition each change within the actually executed context but with reference to the knowledge state of the humans when they formulated the requests. For example, operators may call for removal of a goal that was actually already accomplished, or they may request a loosely targeted goal whose precise location has since been more accurately determined by the rover.

Keeping a consistent vehicle model synchronized among all the components of SRR is an outstanding challenge. While a single activity dictionary serves as the original source for the planner and executive models, regeneration from the source is only automated for the planner model, and even then involves some additional manual tweaks. The locomotion components are fully independent and must be kept in sync manually, e.g. when the rover speed is updated. Fully automated generation of each component's model from a single spacecraft description would be better. This would also allow more direct correspondence between planner-level abstractions with the underlying vehicle states reported by low-level components.

Conclusion

Effective integration of planning and execution components within the Self-Reliant Rover architecture enables the robotic explorer to operate productively for long periods with only high-level guidance from human operators. The rover domain presents many unique demands on such an integrated system, which have been addressed by a range of practical techniques, with varying degrees of complication and success. The system was deployed on the Athena rover and demonstrated within a simulated Mars mission vignette, where it realized significant productivity gains over traditional operations techniques. Despite this achievement, there are still many open avenues for tighter integration among the system's planning and execution components.

Acknowledgments

Portions of this work were performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

References

Chien, S.; Knight, R.; Stechert, A.; Sherwood, R.; and Rabideau, G. 2000. Using iterative repair to improve responsiveness of planning and scheduling. In *International Conference on Artificial Intelligence Planning Systems (AIPS 2000)*.

Edwards, C. D.; Barela, P. R.; Gladden, R. E.; Lee, C. H.; and Paula, R. D. 2014. Replenishing the mars relay network. In *2014 IEEE Aerospace Conference*, 1–13.

Francis, R.; Estlin, T.; Doran, G.; Johnstone, S.; Gaines, D.; Verma, V.; Burl, M.; Frydenvang, J.; Montaño, S.; Wiens, R. C.; Schaffer, S.; Gasnault, O.; DeFlores, L.; Blaney, D.; and Bornstein, B. 2017. Aegis autonomous targeting for chemcam on mars science laboratory: Deployment and results of initial science team use. *Science Robotics* 2(7).

Gaines, D.; Doran, G.; Justice, H.; Rabideau, G.; Schaffer, S.; Verma, V.; Wagstaff, K.; Vasavada, A.; Huffman, W.; Anderson, R.; Mackey, R.; and Estlin, T. 2016. Productivity challenges for Mars rover operations: A case study of Mars Science Laboratory operations. Technical Report D-97908, Jet Propulsion Laboratory.

Gaines, D.; Doran, G.; Justice, H.; Rabideau, G.; Schaffer, S.; Verma, V.; Wagstaff, K.; Vasavada, A.; Huffman, W.; Anderson, R.; Mackey, R.; and Estlin, T. 2017a. A case study of productivity challenges in mars science laboratory operations. In *International Workshop on Planning and Scheduling for Space (IWPSS 2017)*.

Gaines, D.; Rabideau, G.; Doran, G.; Schaffer, S.; Wong, V.; Vasavada, A.; and Anderson, R. 2017b. Expressing campaign intent to increase productivity of planetary exploration rovers. In *International Workshop on Planning and Scheduling for Space (IWPSS 2017)*.

Goldberg, S. B.; Maimone, M. W.; and Matthies, L. 2002. Stereo vision and rover navigation software for planetary exploration. In *Aerospace Conference Proceedings, 2002. IEEE*, volume 5, 5–5. IEEE.

Powell, M. W.; Shams, K. S.; Wallick, M. N.; Norris, J. S.; Joswig, J. C.; Crockett, T. M.; Fox, J. M.; Torres, R. J.; Kurien, J. A.; and McCurdy, M. P. 2009. Mslice science activity planner for the mars science laboratory mission. Technical report, Jet Propulsion Laboratory.

Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; and Ng, A. Y. 2009. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, 5.

Rothrock, B.; Kennedy, R.; Cunningham, C.; Papon, J.; Heverly, M.; and Ono, M. 2016. Spoc: Deep learning-based terrain classification for mars rover missions. In *AIAA SPACE 2016*. 5539.

Vasavada, A.; Grotzinger, J.; Arvidson, R.; Calef, F.; Crisp, J.; Gupta, S.; Hurowitz, J.; Mangold, N.; Maurice, S.; Schmidt, M.; et al. 2014. Overview of the mars science laboratory mission: Bradbury landing to yellowknife bay and beyond. *Journal of Geophysical Research: Planets* 119(6):1134–1161.

Verma, V.; Gaines, D.; Rabideau, G.; Schaffer, S.; and Joshi, R. 2017. Autonomous science restart for the planned europa mission with lightweight planning and execution. In *International Workshop on Planning and Scheduling for Space (IWPSS 2017)*.

Volpe, R.; Nesnas, I.; Estlin, T.; Mutz, D.; Petras, R.; and Das, H. 2001. The claraty architecture for robotic autonomy. In *Aerospace Conference, 2001, IEEE Proceedings.*, volume 1, 1–121. IEEE.

Weiss, K. 2013. An introduction to the jpl flight software product line. In *Proceedings of the 2013 Workshop on Spacecraft Flight Software (FSW-13)*.